**CROP RECOMMENDATION SYSTEM**

CAPSTONE PROJECT

Submitted in partial fulfillment of the requirements of the

**Post Graduate Certification Program**

**in**

**Artificial Intelligence and Machine Learning**

By

**IMTIYAZ ALAMSHAH 2024AIML132**

**TEJASWINI LALITHA 2024AIML587**

**ANKIT MISHRA 2024AIML509**

**BAGEPALLI SOUMYASREE 2024AIML606**

Under the supervision of

**Prof. Aniruddha Dasgupta**

Project work carried out at

**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE**

**Pilani (Rajasthan) INDIA**

(October,2025)

**PGCP AIML CAPSTONE PROJECT**

**CROP RECOMMENDATION SYSTEM**

Submitted in partial fulfillment of the requirements of the

PGCP - Artificial Intelligence and Machine Learning

By

**IMTIAYZ ALAMSHAH 2024AIML132**

**LIPSA MISHRA 2024AIML067**

**RAIBAGKAR HARSHAL GOPALRAO 2024AIML057**

**SHAH HIMA KIRANBHAI 2024AIML011**

**UPADYAYULA V SHARMA 2024AIML070**

Under the supervision of

**Prof. Aniruddha Dasgupta**

Project work carried out at

**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE**

**PILANI (RAJASTHAN)**

(October, 2025)

**ACKNOWLEDGEMENTS**

We express our deep sense of gratitude to BITS, Pilani for providing an opportunity to pursue our project on “**Crop Recommendation System**”.

We thank our mentor, **Prof. Aniruddha Dasgupta** for the constant support and guidance which resulted in the successful completion of the project within the specified time. His unflinching help and encouragement were a constant source of inspiration to us.

We would also like to thank all our professors, who taught us the basic and advanced concepts of artificial intelligence and machine learning in such a nuanced manner, during this certification program.

We would also like to thank our respective families and organizations for supporting us through this journey.

**IMTIAYZ ALAMSHAH (2024AIML132)**

**LIPSA MISHRA (2024AIML067)**

**RAIBAGKAR HARSHAL GOPALRAO (2024AIML057)**

**SHAH HIMA KIRANBHAI (2024AIML011)**

**UPADYAYULA V SHARMA (2024AIML070)**

**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI**

**CERTIFICATE**

This is to certify that the Capstone Project entitled “**Crop Recommendation System**”

and submitted by **IMTIAYZ ALAMSHAH (2024AIML132), LIPSA MISHRA (2024AIML067), RAIBAGKAR HARSHAL GOPALRAO (2024AIML057), SHAH HIMA KIRANBHAI (2024AIML011) AND UPADYAYULA V SHARMA (2024AIML070)** in partial fulfillment of the requirements of PGCP AIML Capstone Project, embodies the work done by him/her under my supervision.

Place : Hyderabad Signature of the Mentor

Date : 20-10-2025 Name : Prof. Aniruddha Dasgupta

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI**

**2024-25**

**PGCP AIML CAPSTONE PROJECT**

Project Title : Crop Recommendation System

Name of Mentor : Prof. Aniruddha Dasgupta

Name of Student : IMTIAYZ ALAMSHAH, LIPSA MISHRA, RAIBAGKAR HARSHAL GOPALRAO, SHAH HIMA KIRANBHAI, UPADYAYULA V SHARMA

ID No. of Student : 2024AIML132, 2024AIML067, 2024AIML057, 2024AIML011, 2024AIML070

## **ABSTRACT**

Agriculture is the backbone of India’s economy, supporting a majority of the rural population. However, farmers often face challenges in selecting the most suitable crop due to variations in soil type, climatic conditions, and lack of technical knowledge. The **Crop Recommendation System in India** addresses this issue by providing a scientific and data-driven approach to crop selection. It leverages agricultural and environmental data to assist farmers in making informed decisions that can enhance productivity and sustainability.

The system uses machine learning algorithms to analyze parameters such as soil type, pH level, nutrient composition, temperature, and rainfall. Based on these inputs, it predicts the most appropriate crop for a specific region. By integrating technology with agriculture, the project promotes efficient resource utilization, improved crop yield, and long-term food security. This initiative demonstrates the potential of artificial intelligence in revolutionizing traditional farming practices in India.

.

## **TABLE OF CONTENTS**

[Problem Statement 7](#_ik3p3tyg8xd9)

[Objective of the Project 7](#_gmge5zlth7bj)

[Machine Learning Process Flow (Architecture) 8](#_vjy1qobjakpm)

[Resources or Prerequisite 8](#_9u20vf5f4fyv)

[Potential data challenges & risks in doing the project 9](#_2079qjstd4m0)

[Detailed Plan of Work 10](#_fvx49uizcv5i)

[Dataset 11](#_3pruo97ks3r3)

[Pre-Processing and Feature Engineering 12](#_wb5ox9kngkzj)

[Machine Learning Modeling & Techniques Applied 17](#_etd7qhipll84)

Objective: 17

Modeling Approach: 17

[Models Trained: 17](#_izgnf2gyux9y)

[Training Details: 17](#_td2k1y3ehutk)

[Code & Screenshots 18](#_ejbim3i30hi8)

[Deployment 28](#_x1q8dzt9fbsu)

Objective: 28

Tools and Technologies Used: 28

[Deployment Architecture Overview: 28](#_7qhdh7m5aeb7)

[Project Folder Structure: 28](#_p0n3u9sotzmi)

[Key Insights & Impactful/Important Features 29](#_f21jv3aly7ut)

[Key Insights 29](#_bx96rvci6o30)

[Important Features Identified 29](#_9xtqsvfrfkxm)

[Future Work / Scope of improvements 30](#_s1wqmwd1efjl)

[1. Expansion of Historical Data 30](#_a2cqskye1s6m)

[2. Incorporating More Features 30](#_1mwdb23hez6x)

[BIBLIOGRAPHY 31](#_10gvsajen0il)

[Checklist of items for the Final report 31](#_jgnjez5pipxl)

## 

## **Problem Statement**

Agriculture in India is highly dependent on environmental conditions and soil characteristics, which vary widely across regions. Most farmers rely on traditional knowledge or past experience to decide which crop to cultivate, often without access to scientific data or expert guidance. This leads to suboptimal crop choices, reduced productivity, and financial losses.

Unpredictable weather patterns, irregular rainfall, and soil degradation further complicate the decision-making process. Farmers are frequently unaware of how soil nutrients, pH level, or moisture content affect crop suitability. Consequently, they may invest time and resources into crops that are not well-suited to their land or local climate.

Therefore, there is a pressing need for an intelligent, data-driven system that can recommend the most suitable crop for a given location based on real-time agricultural parameters. Such a system can help optimize yields, ensure sustainable farming, and support better livelihood outcomes for Indian farmers.

## **Objective of the Project**

## The objective of the **Crop Recommendation System in India** is to design and implement an intelligent, data-driven application capable of predicting the most suitable crop for a given region using machine learning and environmental analysis. The project focuses on building a technically robust solution that processes agricultural data efficiently and provides accurate, actionable insights to farmers.

## **Odbjectives include:**

## To collect and preprocess agricultural datasets containing soil, weather, and crop parameters.

## To perform data cleaning, normalization, and feature extraction for model training.

## To implement and compare machine learning algorithms such as Random Forest, Decision Tree, and Support Vector Machine for crop prediction.

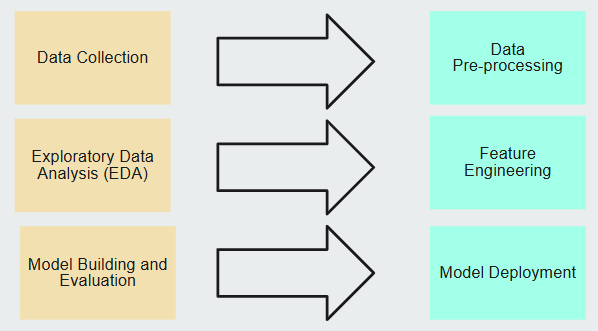
## To evaluate model accuracy using appropriate metrics like precision, recall, and F1-score.

## To develop a web or desktop-based user interface for inputting soil and climate data.

## To integrate the trained model with the interface to generate real-time crop recommendations.

## To deploy the system for testing and validate its performance using real-world agricultural data.

## **Machine Learning Process Flow (Architecture)**

****

## **Resources or Prerequisite**

Data Resources: Collect data from various sources <https://etrain.info/in>, <https://indianrail.gov.in>

Libraries & Frameworks:

* *pandas, numpy* for data manipulation
* *matplotlib, seaborn* for visualization
* *scikit-learn, xgboost* for machine learning
* *Streamlit* for deployment

Software: Python

## **Potential data challenges & risks in doing the project**

Developing a **Crop Recommendation System** involves handling complex and diverse agricultural datasets, which presents several data-related challenges and risks. The accuracy and effectiveness of the model depend heavily on the quality, completeness, and consistency of data used for training and testing.

One major challenge is the **availability and accessibility of reliable agricultural data**. Data on soil nutrients, weather conditions, and crop yields are often scattered across multiple government and private sources, making collection and integration difficult. Additionally, **data inconsistency and missing values** can impact the performance of machine learning algorithms, requiring extensive preprocessing and cleaning.

Another risk is **regional data imbalance**, where certain states or soil types are overrepresented, leading to biased predictions. **Temporal changes** in climate patterns and soil fertility can also make historical data less relevant over time. Finally, ensuring **data privacy and security** is critical, especially when integrating data from farmers or government databases. Addressing these challenges is essential to ensure the system’s accuracy, scalability, and long-term reliability.

## **Proposed Methodology**

The **Crop Recommendation System in India** aims to utilize data analytics and machine learning techniques to predict the most suitable crop for a given region based on soil and environmental parameters. The proposed methodology is divided into systematic stages that ensure efficient data handling, model training, and user interaction.

1. **Data Collection:**  
   Relevant datasets are collected from credible sources such as the data.gov.in, Kaggle data source, and government open data portals. The datasets include soil characteristics (pH, nitrogen, phosphorus, potassium), weather parameters (temperature, rainfall), and regional crop yield data.
2. **Data Preprocessing:**  
   Collected data undergo cleaning to remove missing, duplicate, or inconsistent entries. Feature scaling and normalization are applied to ensure uniformity. Outliers are identified and treated to improve model stability.
3. **Model Development:**  
   Various machine learning algorithms—such as **Random Forest**, **Decision Tree**, **Support Vector Machine (SVM)**, and **K-Nearest Neighbor (KNN)**—are trained and tested on the processed data. The models are tuned using hyperparameter optimization techniques to enhance prediction accuracy.
4. **Model Evaluation:**  
   Performance is assessed using metrics like accuracy, precision, recall, F1-score, and confusion matrix. Cross-validation techniques are applied to ensure the model’s generalization and reliability.
5. **Testing and Validation:**  
   The system is tested using real-time or region-specific data to validate the accuracy of recommendations. Feedback from agricultural experts and field trials can further refine the model.
6. **Deployment and Future Enhancement:**  
   After validation, the system is deployed for use by farmers and agricultural departments. Future enhancements may include IoT-based sensor integration, real-time weather APIs, and mobile application development for wider accessibility.

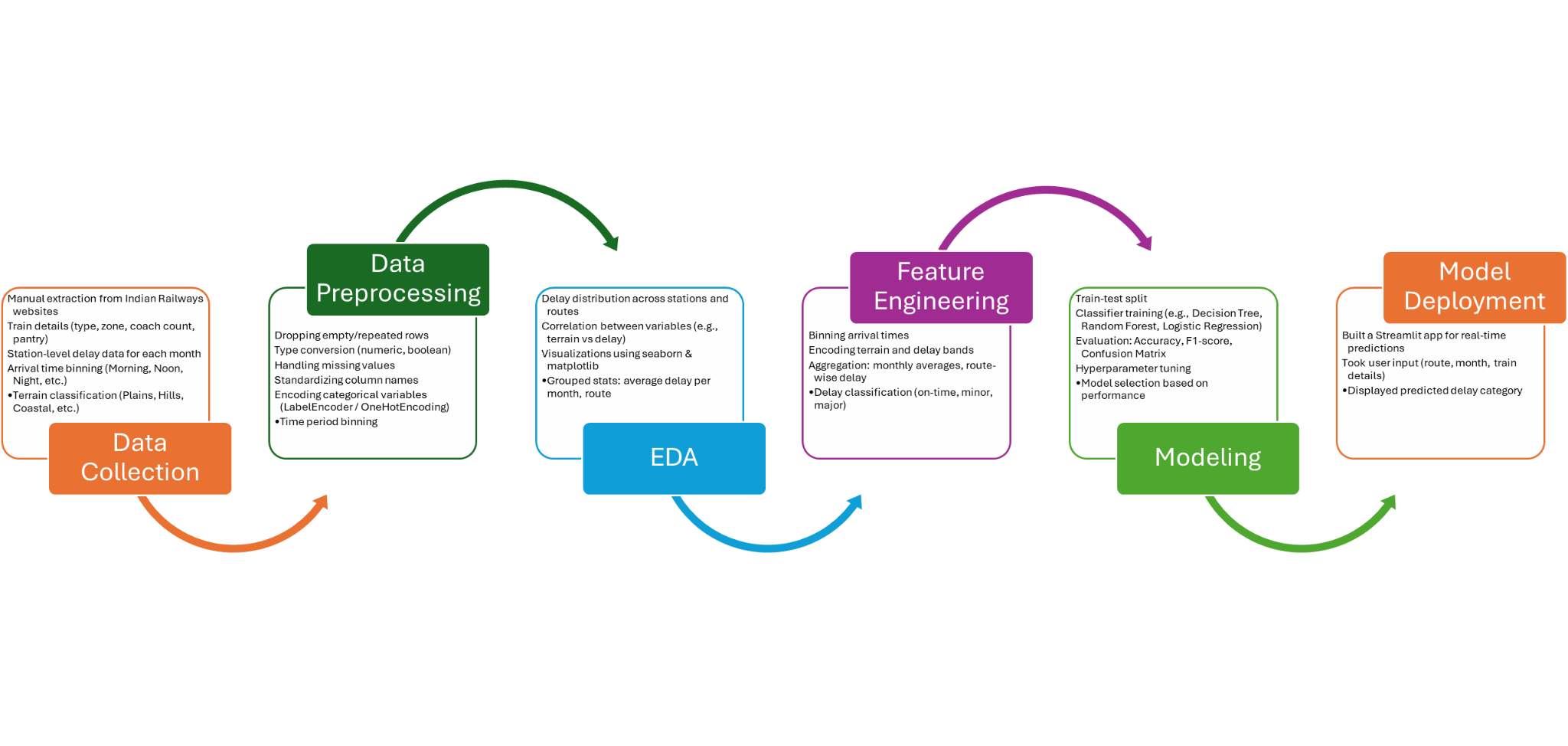
## **Detailed Plan of Work**

*Week 1-2: Data exploration and preprocessing.*

*Week 3-4: Feature engineering and EDA.*

*Week 5-6: Model training and hyperparameter tuning.*

*Week 7-8: Evaluation, deployment, and documentation*

****

**Problem Definition:** Identified the need to predict train delays (Low, Moderate, High) for trains arriving at Delhi from various cities to support operational planning.

**Data Collection:** Collected train and station data manually from [indianrail.gov.in](http://indianrail.gov.in) and [etrain.info](https://etrain.info/in) for selected routes (e.g., Bangalore–Delhi, Pune–Delhi).  
 Monthly delays were extracted by visually interpreting graphs and estimating (min, max, mode) delays.

**Data Preprocessing** Cleaned and standardized data, handled missing values, encoded categorical features (e.g., terrain, time slots).

**Exploratory Data Analysis (EDA)** Visualized delay patterns across train types, terrains, and time slots to identify key influencing factors.

**Feature Engineering** Created derived features such as terrain complexity, travel time, and monthly delay spread.

**Model Building and Evaluation** Trained multiple classification models (Random Forest, XGBoost, etc.).  
 Evaluated using accuracy, F1-score, and confusion matrix to select the best-performing model.

**Model Deployment** Deployed the final model using **Streamlit**, allowing users to input train details and get delay category predictions in real-time.

## **Dataset**

The dataset used in this project comprises two main components:

**1. Primary Train Metadata Sheet** This sheet contains high-level attributes for each train considered in the study. Each row represents a unique train route, with features that describe both operational and scheduling details. The key columns include:

* **Train Number and Name**: Unique identifiers for each train.
* **Train Type and Zone**: Indicates the service category (e.g., Superfast, Rajdhani, Duronto) and administrative railway zone (e.g., SWR, NR).
* **Coach Count and Pantry Availability**: Represents the total number of coaches and whether a pantry car is present.
* **Operational Days**: A binary indicator for each day of the week specifying whether the train runs on that day.
* **Route Details**: Including origin, destination, departure/arrival times, and total travel time.
* **Classes Available**: Lists all travel classes (e.g., 1A, 2A, 3A, SL) offered on the route.

This metadata sheet forms the basis for high-level features used in machine learning modeling, including encoding of categorical variables, travel time calculation, and operational patterns.

**2. Individual Station-wise Delay Sheets** Each train has a dedicated sheet listing its route in sequential station order. Each station entry provides both structural and delay information:

* **Station Code and Name**: Identifies the stations on the train’s route.
* **Station Number**: The sequential order of stations for the route.
* **Terrain Type**: Inferred category (e.g., Plains, Plateau, Coastal) to assess geographical influence on delays.
* **Number of Platforms and Distance**: Indicates infrastructure capacity and cumulative distance from the origin.
* **Arrival Time**: Scheduled arrival at that station.
* **Monthly Delay Metrics**: For each month (April to March), delay values are recorded as a tuple (Min, Max, Mode), representing estimated arrival delay ranges at that station.

Delays were manually extracted and approximated from graphical plots available on public railway portals (e.g., etrain.info). This structure enabled the derivation of both numerical and categorical features relevant to delay behavior, such as average mode delay, seasonal variability, and terrain-based delay patterns

## **Pre-Processing and Feature Engineering**

### **Summary**

### The dataset consists of train-wise route sheets and a master metadata sheet. Each train sheet includes monthly delay metrics at various stations and features such as platform count, terrain, and distance. Preprocessing was performed in two stages:

### **Model training stage** – to clean and transform raw Excel data into a structured processed\_df.

### **App runtime stage** – via preprocess\_input(df) to ensure user input is processed identically.

### **Data Type Correction**

1. Renamed ambiguous or unnamed columns in the metadata sheet.
2. Converted Departure Time and Arrival Time from string to datetime.time.
3. Converted Date column to pandas datetime.date.
4. In the deployed Streamlit app, time\_input and date\_input widgets capture time and date, which are then formatted into strings before processing.

### **To Classify Time Data into Categories or Bins**

1. Extracted the **hour component** from Departure Time and Arrival Time.
2. Binned the hours into four categories:
   1. Morning (5 AM to 12 PM)
   2. Afternoon (12 PM to 5 PM)
   3. Evening (5 PM to 9 PM)
   4. Night (9 PM to 5 AM)
3. Created two new features: Dep\_Hour\_Bin and Arr\_Hour\_Bin.
4. Binning logic is implemented in both model training and preprocess\_input(df) for consistency.

**Handling Missing Values**

For training:

1. Used **mean imputation** for numerical columns such as Travel Time, Average Platform Count, and Monthly Delay.
2. Applied SimpleImputer(strategy='mean') specifically for models like Logistic Regression and SVM, which cannot handle NaNs.
3. Missing terrain values were either filled with 'Unknown' or left unencoded if unused.
4. In the deployed app, user input is clean or gets transformed via imputation when needed.

### **Outlier Detection**

### Outliers were assessed visually using:

### **Histograms** of Average Mode Delay

### **Boxplots** for each month's delay distribution

### Outliers were **not removed** aggressively, as the dataset was small and retaining variability was more important for model generalization.

### **Encoding**

### Encoding was applied as follows:

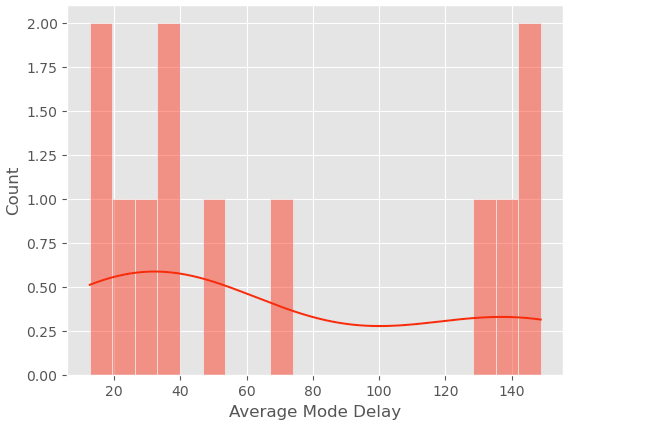
1. **Label Encoding** was used for the target variable: Delay Category.
   1. **One-Hot Encoding** was applied to:
      1. Train Type
      2. Railway Zone
      3. Days of Run
      4. Classes
      5. Dep\_Hour\_Bin and Arr\_Hour\_Bin
   2. **Multi-Label Binarization** was applied for terrain types (Terrain Encountered).
2. All non-numeric features were encoded during training.
3. The same encoding logic is mirrored in the deployment function (preprocess\_input(df)) to ensure consistency during inference.

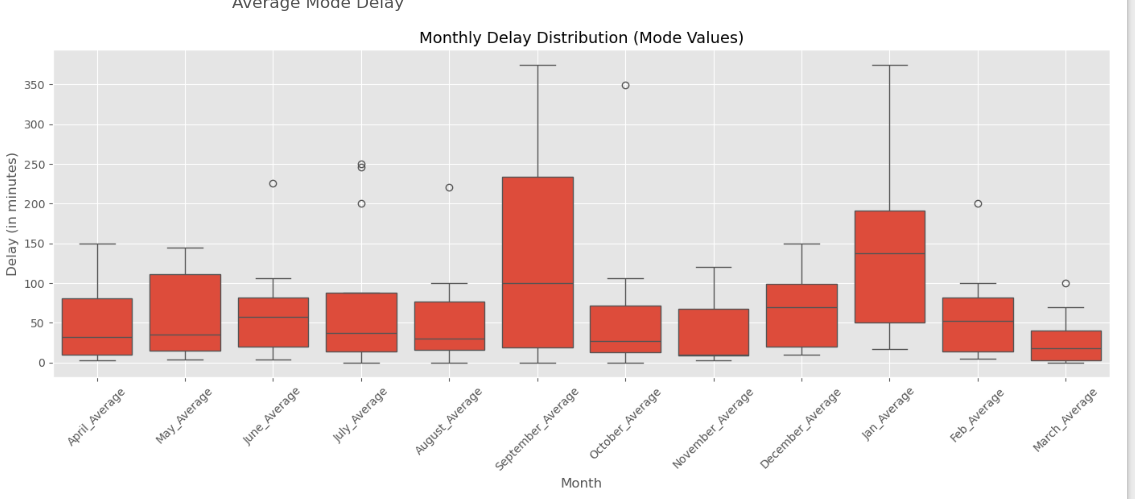
# **Exploratory Data Analysis (EDA) Summary**

## **Overall Objectives**

The goal of the EDA was to understand:  
- The distribution of delays across trains and months  
- Variability in terrain, platform availability, and route length  
- Potential correlations between features like station count, coach type, and delay behavior

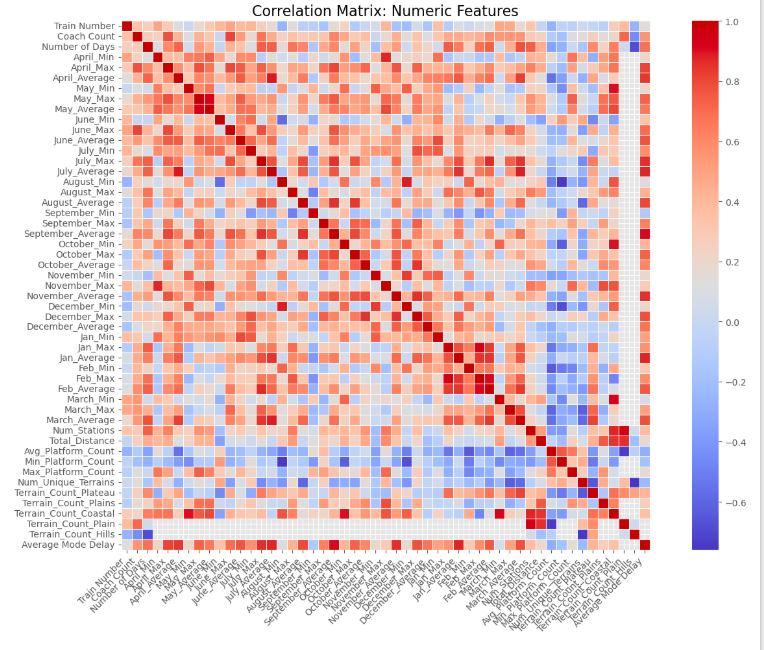
## **Target Variable: Average Mode Delay**



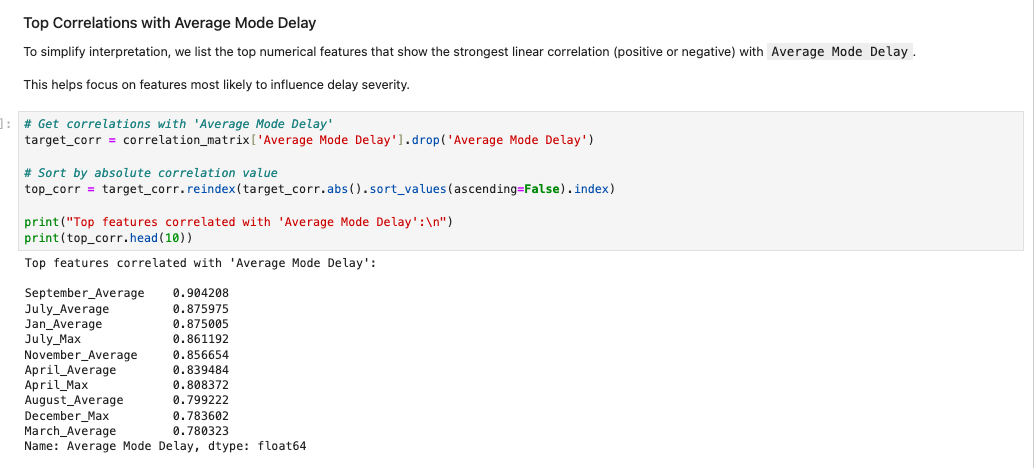
**Monthly Delay Distributions**

**Correlation Analysis**

Correlation Between Features:



Correlation between Target Variable and Features:



## **Machine Learning Modeling & Techniques Applied**

## **Objective:**

The goal was to build a robust classification model that can predict the delay category (High, Medium, Very High) for trains based on journey metadata and route-level features. We implemented and evaluated multiple models to benchmark performance.

## **Modeling Approach:**

We followed a consistent pipeline for all models:  
1. Preprocessed and engineered features using processed\_df  
2. Encoded categorical variables and binned numerical ones  
3. Binned Average Mode Delay into 3 categories (target class)  
4. Used train\_test\_split() with stratification for fair model evaluation  
5. Evaluated using metrics like Accuracy, Confusion Matrix, and F1-score  
6. Saved each model and its corresponding input column structure and imputer (if needed)

## 

## 

## 

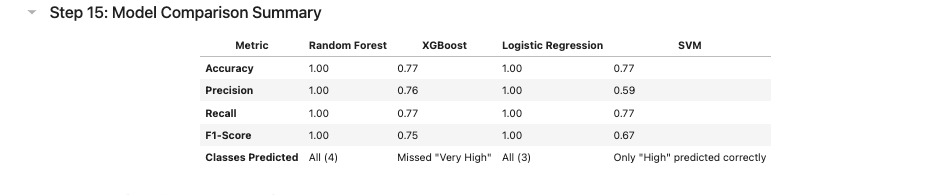
## **Models Trained:**

|  |  |
| --- | --- |
| Model | Notes |
| Random Forest | Baseline, robust to missing data, good performance on tabular data |
| XGBoost | Gradient boosting, strong generalization, best fit for structured data |
| Logistic Regression | Linear classifier, interpretable, required missing value imputation |
| SVM (RBF Kernel) | Non-linear margin classifier, struggled with small dataset |

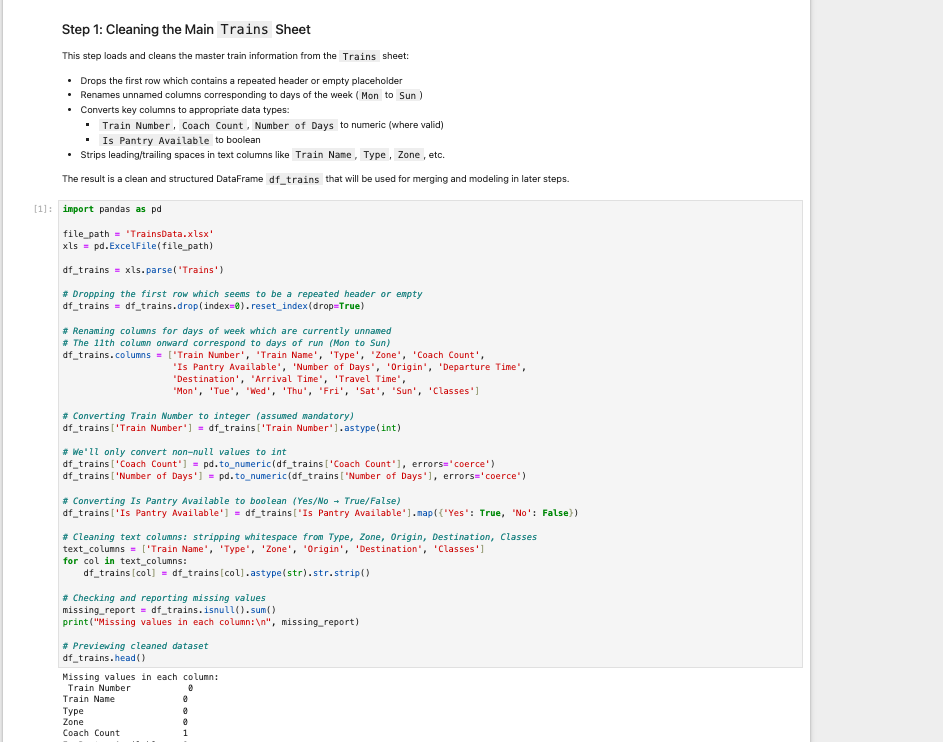
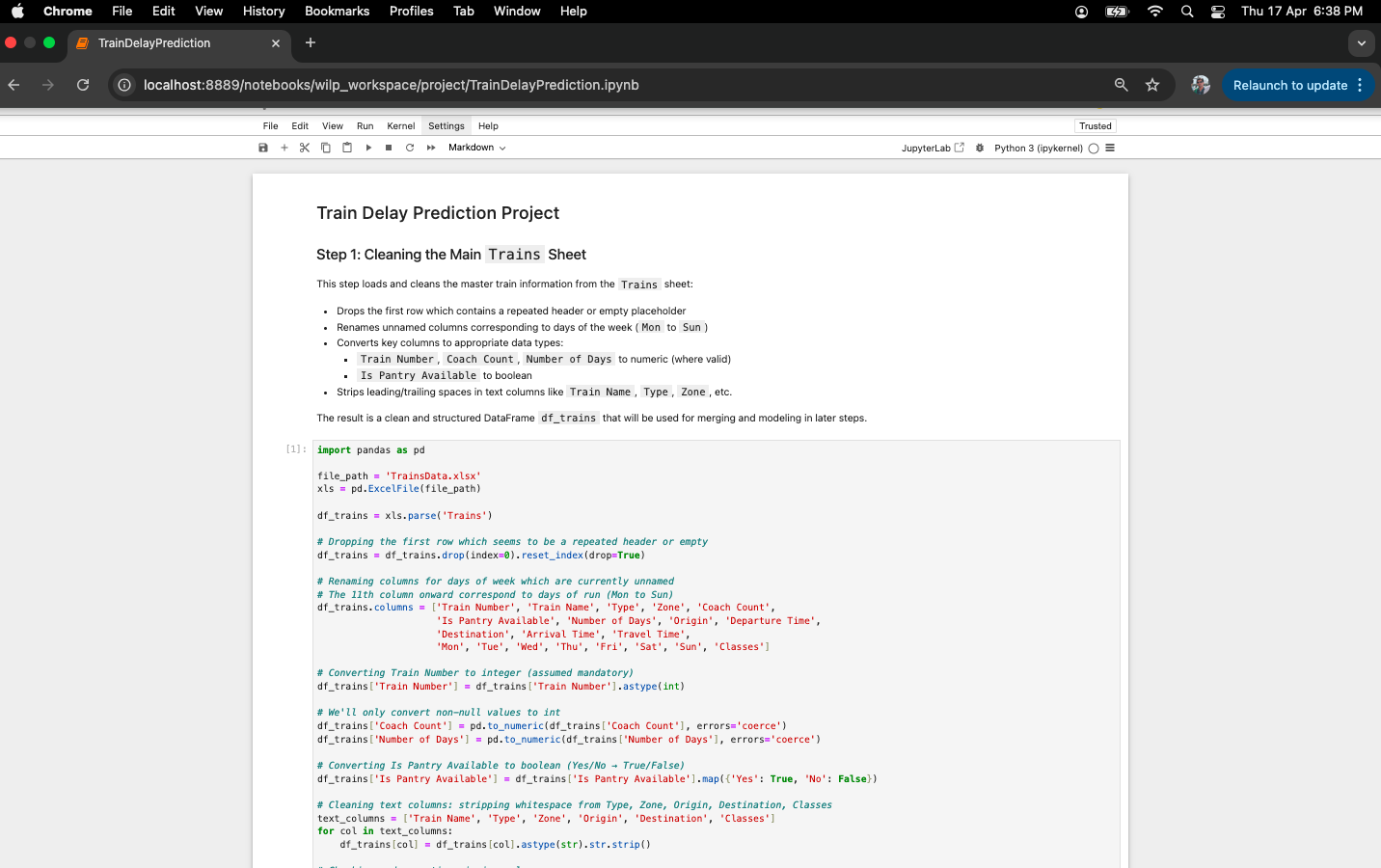
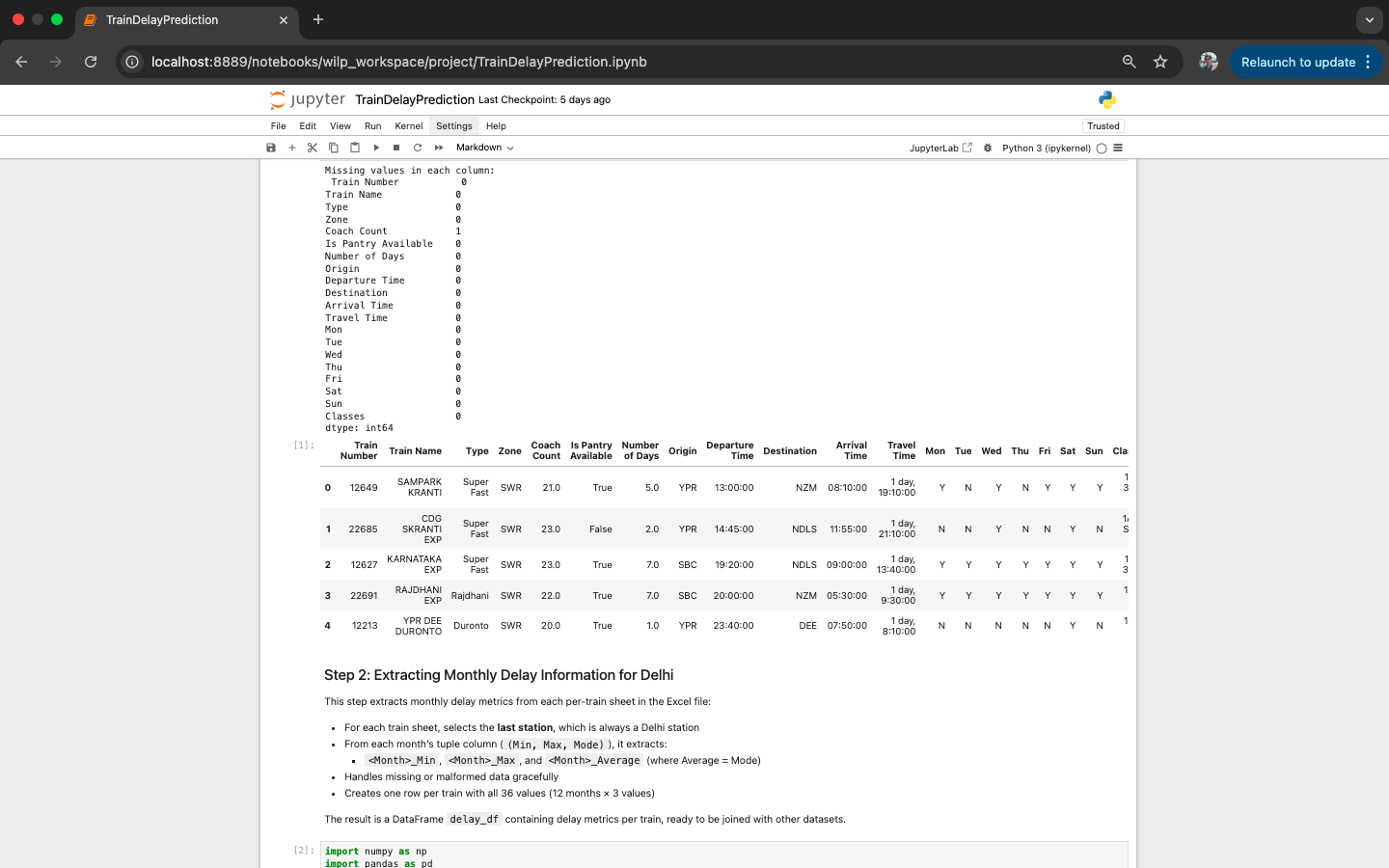
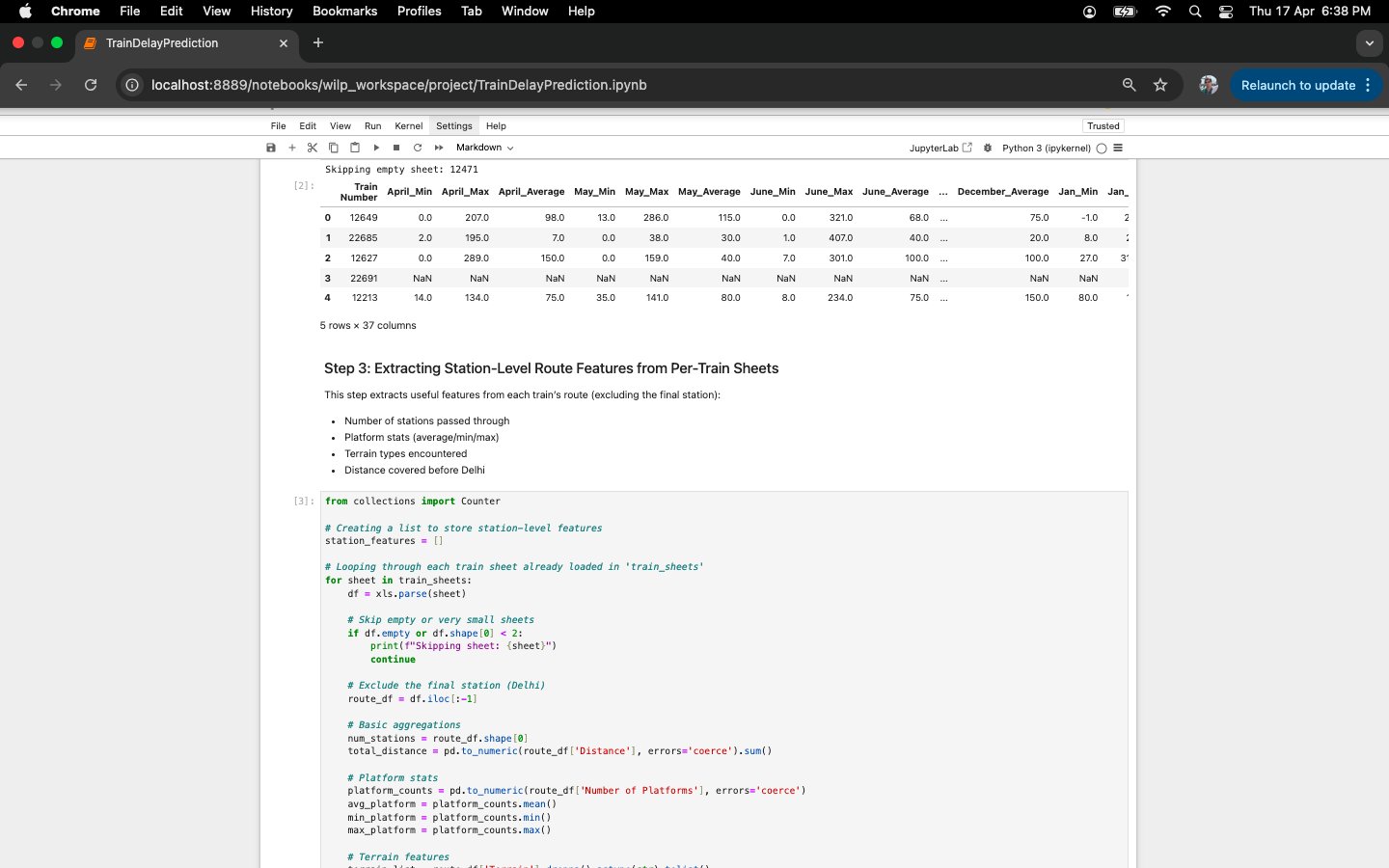
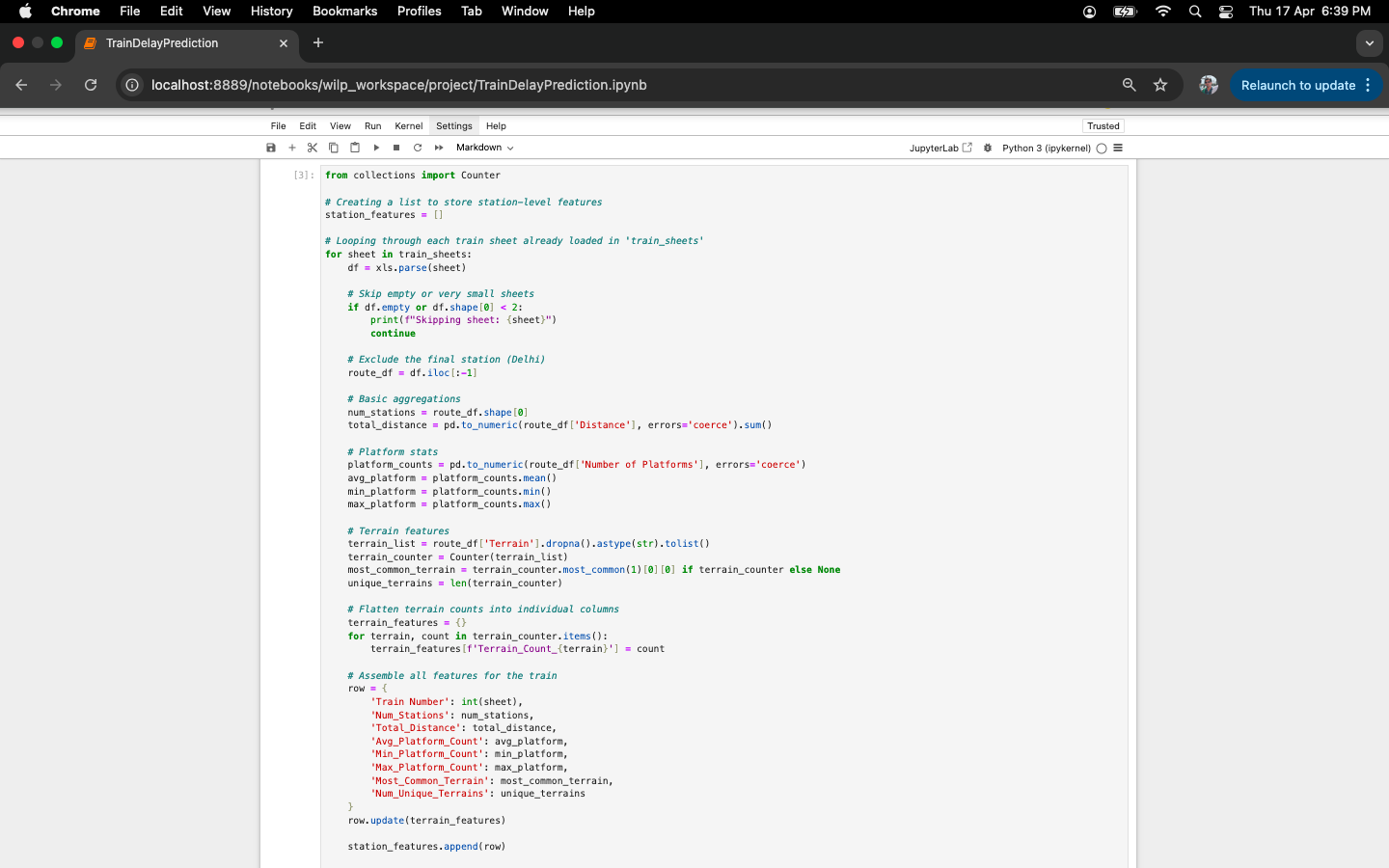
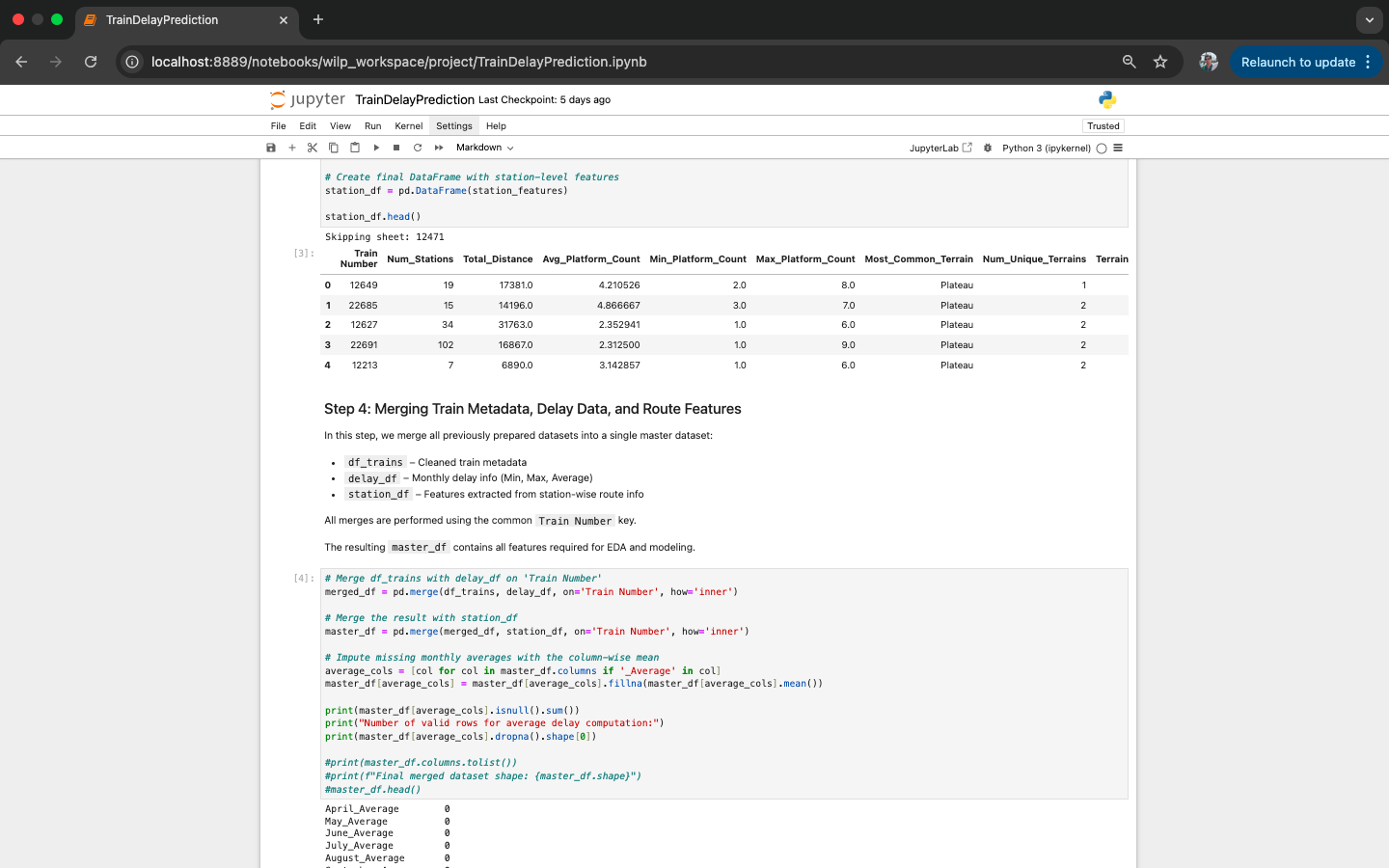
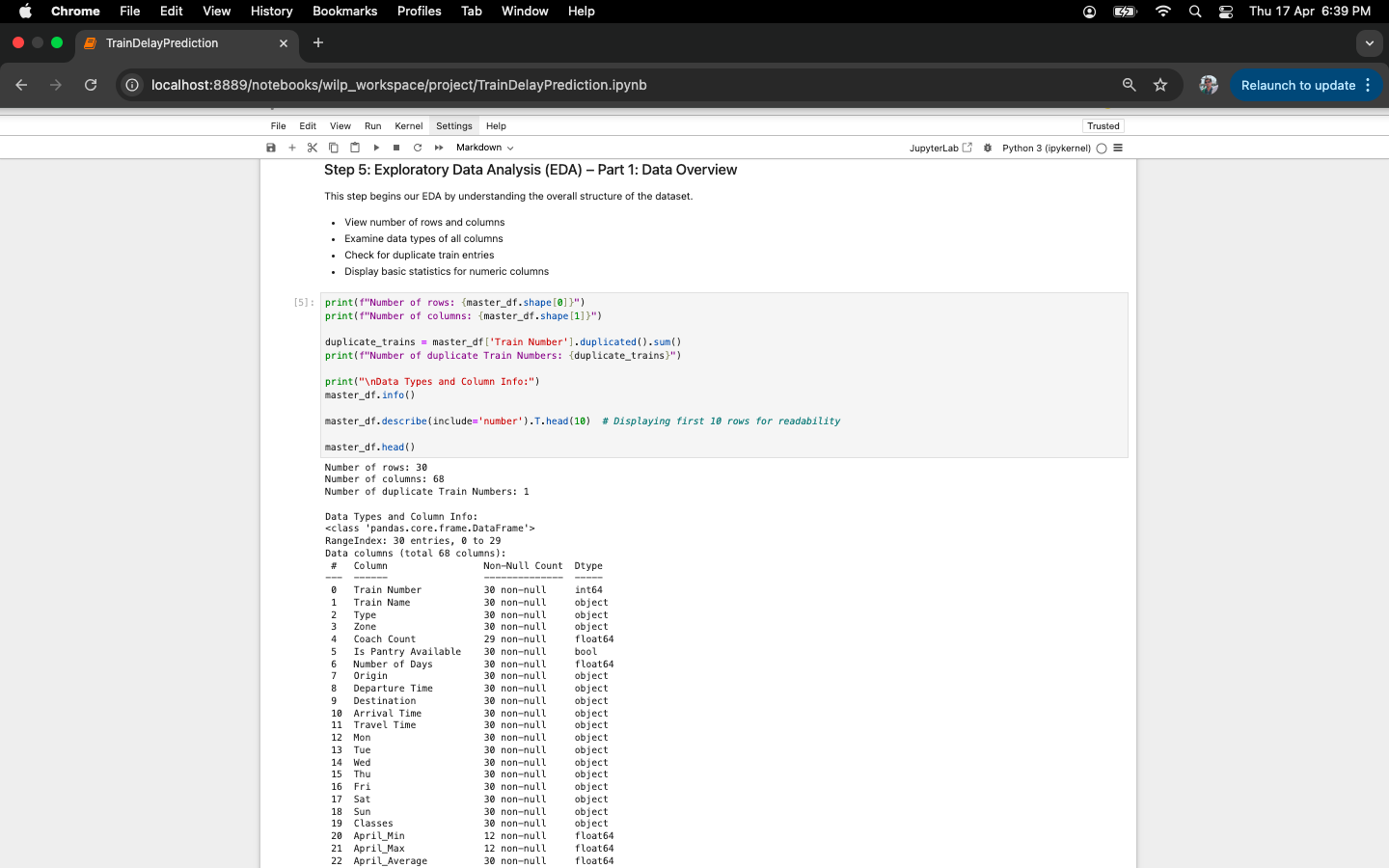
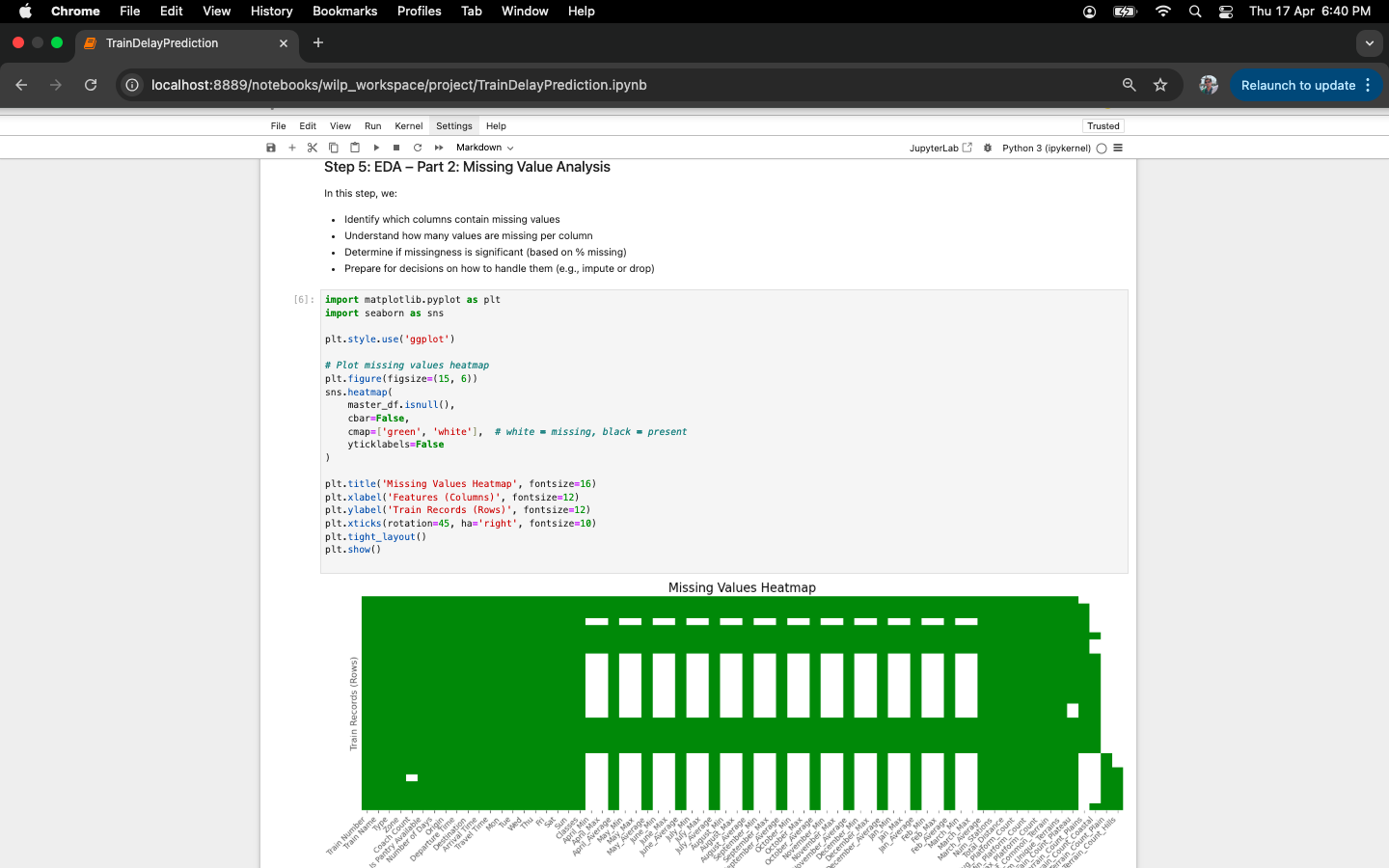
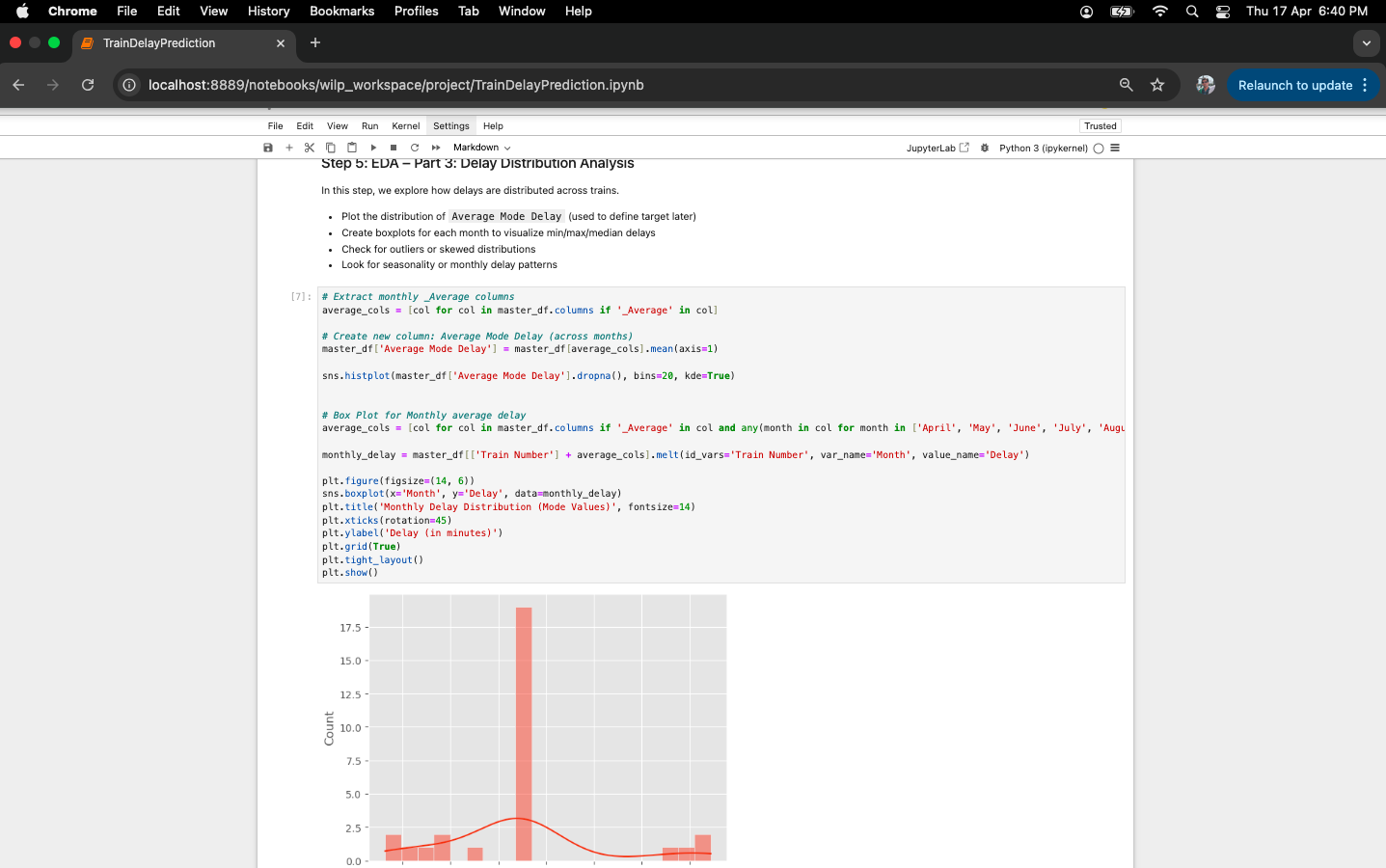
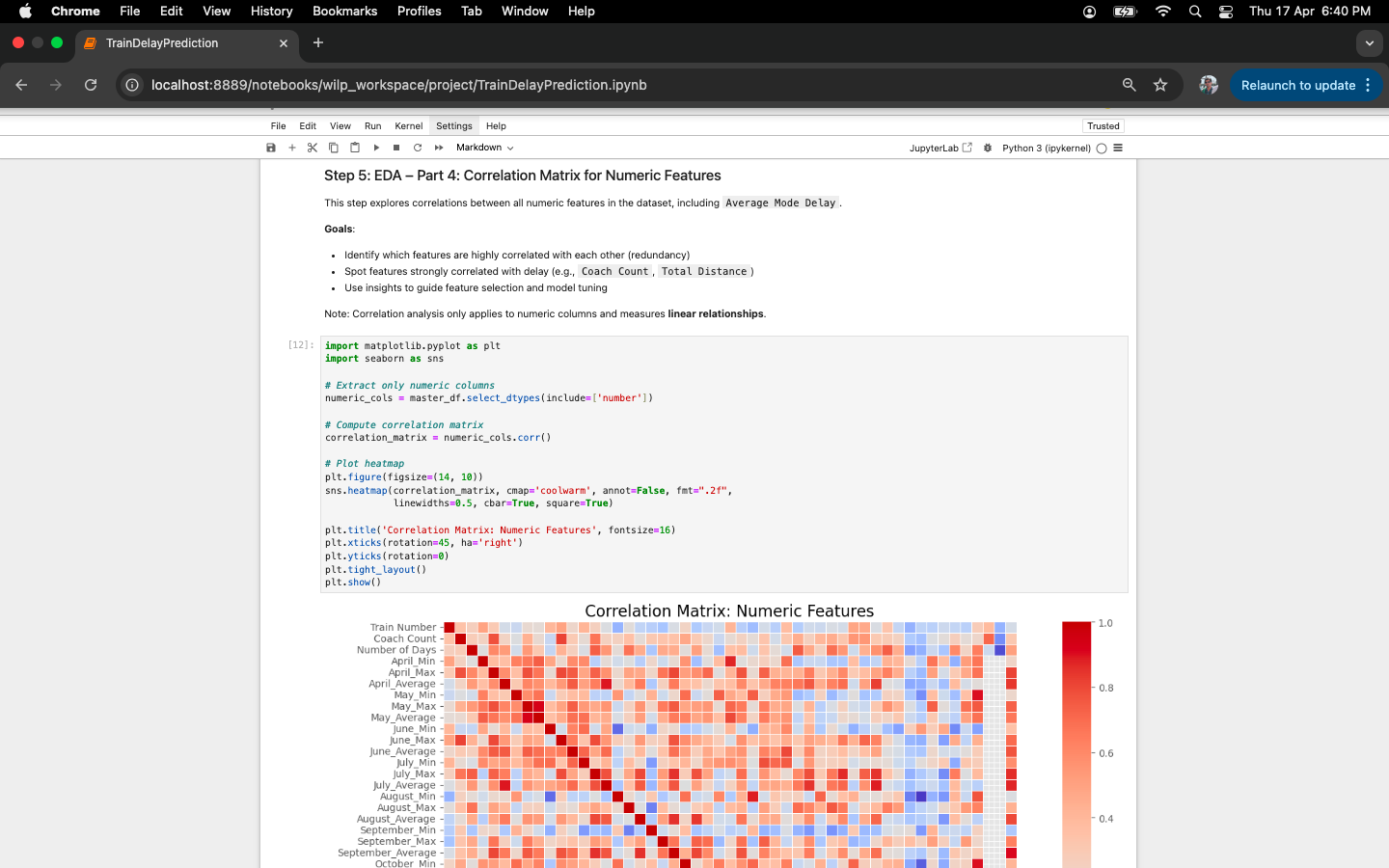
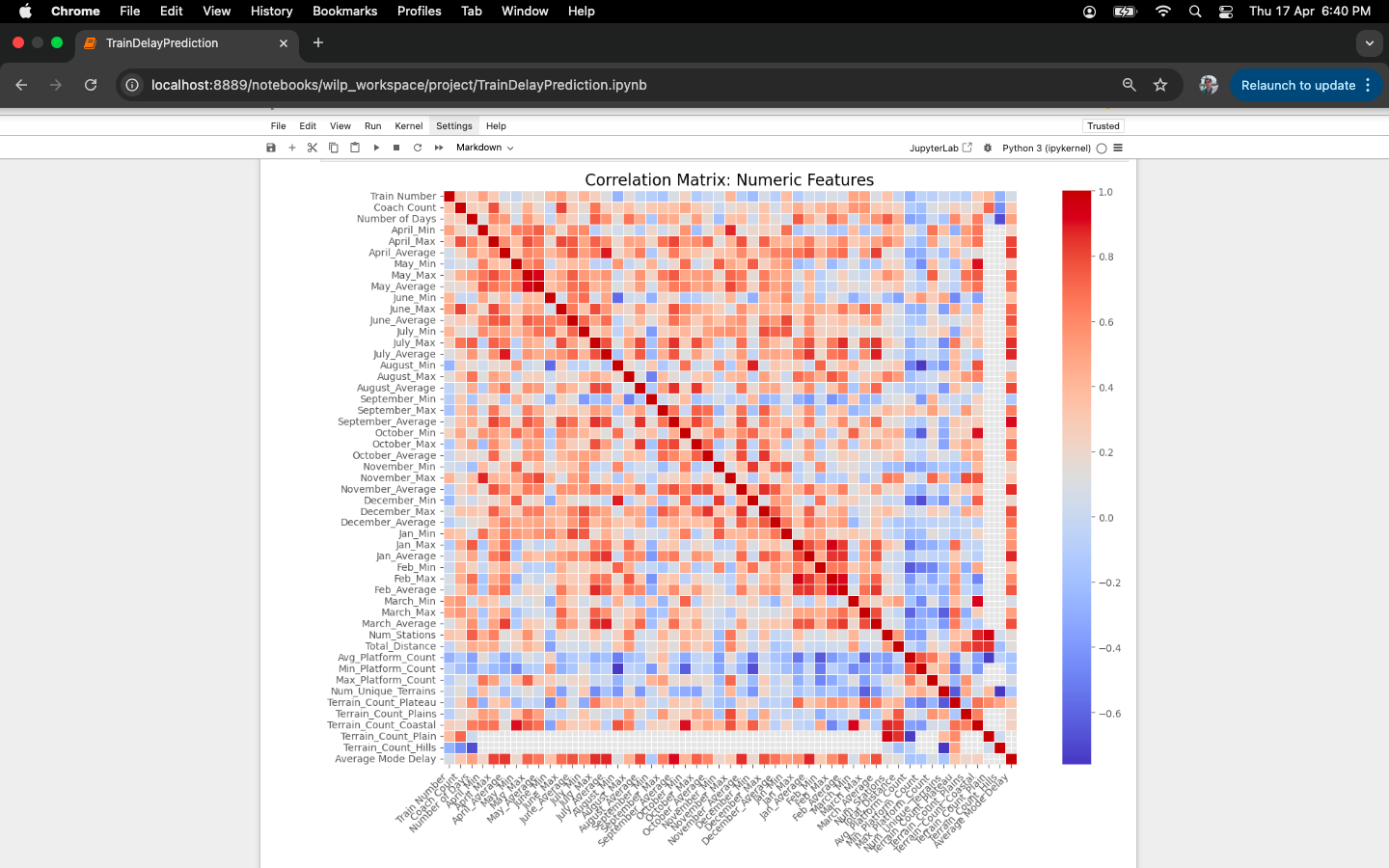
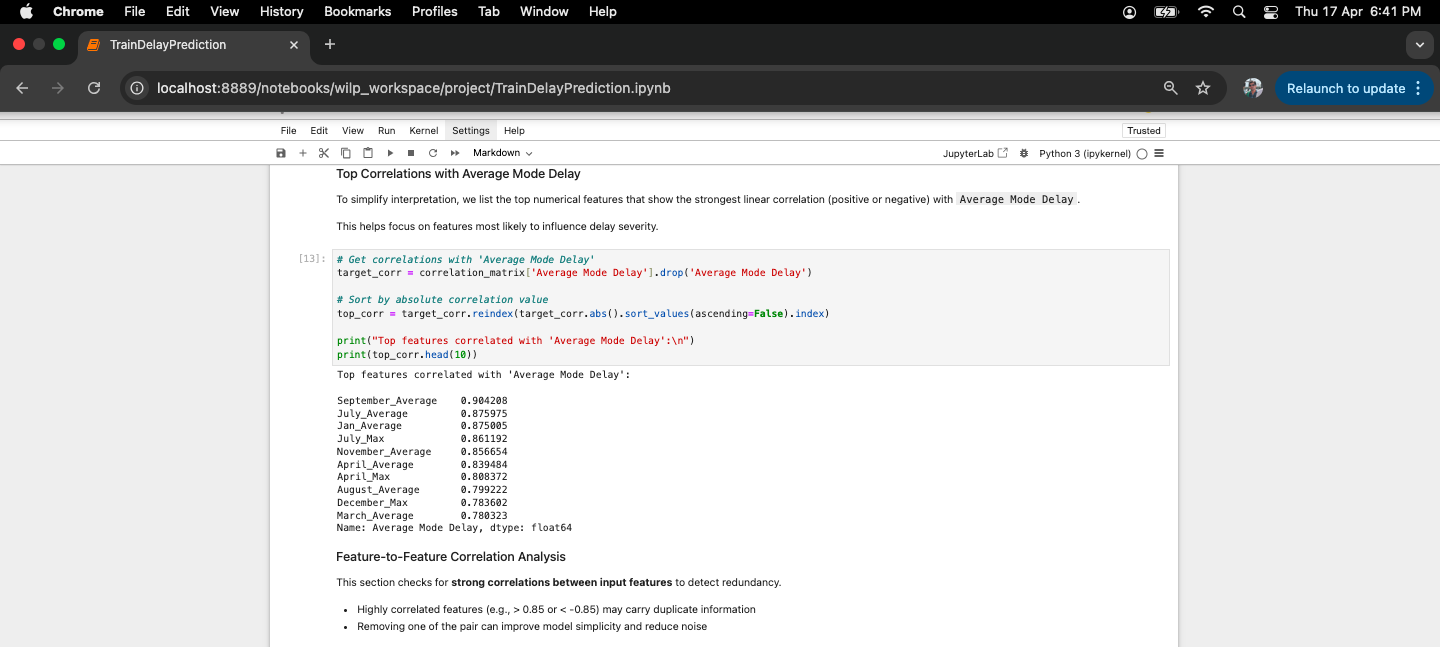
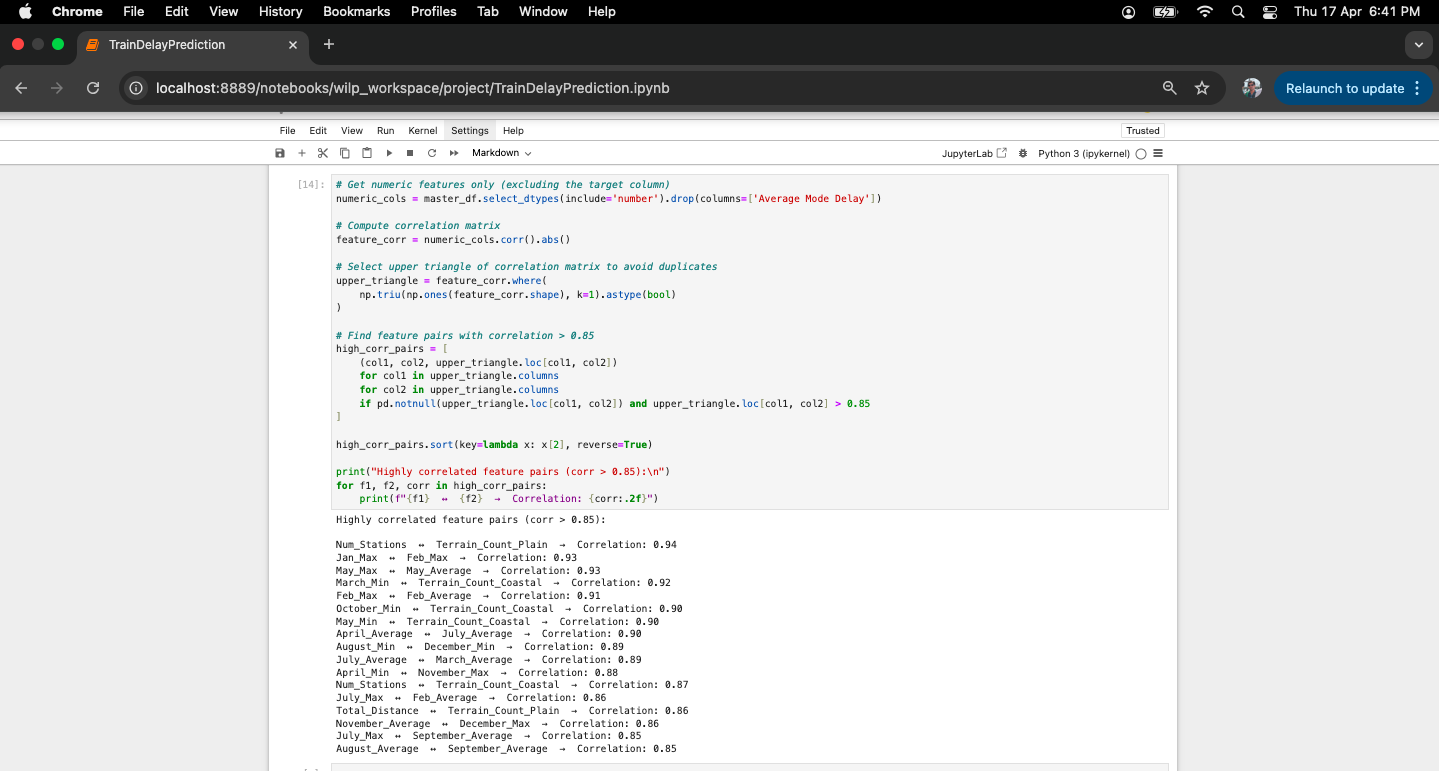
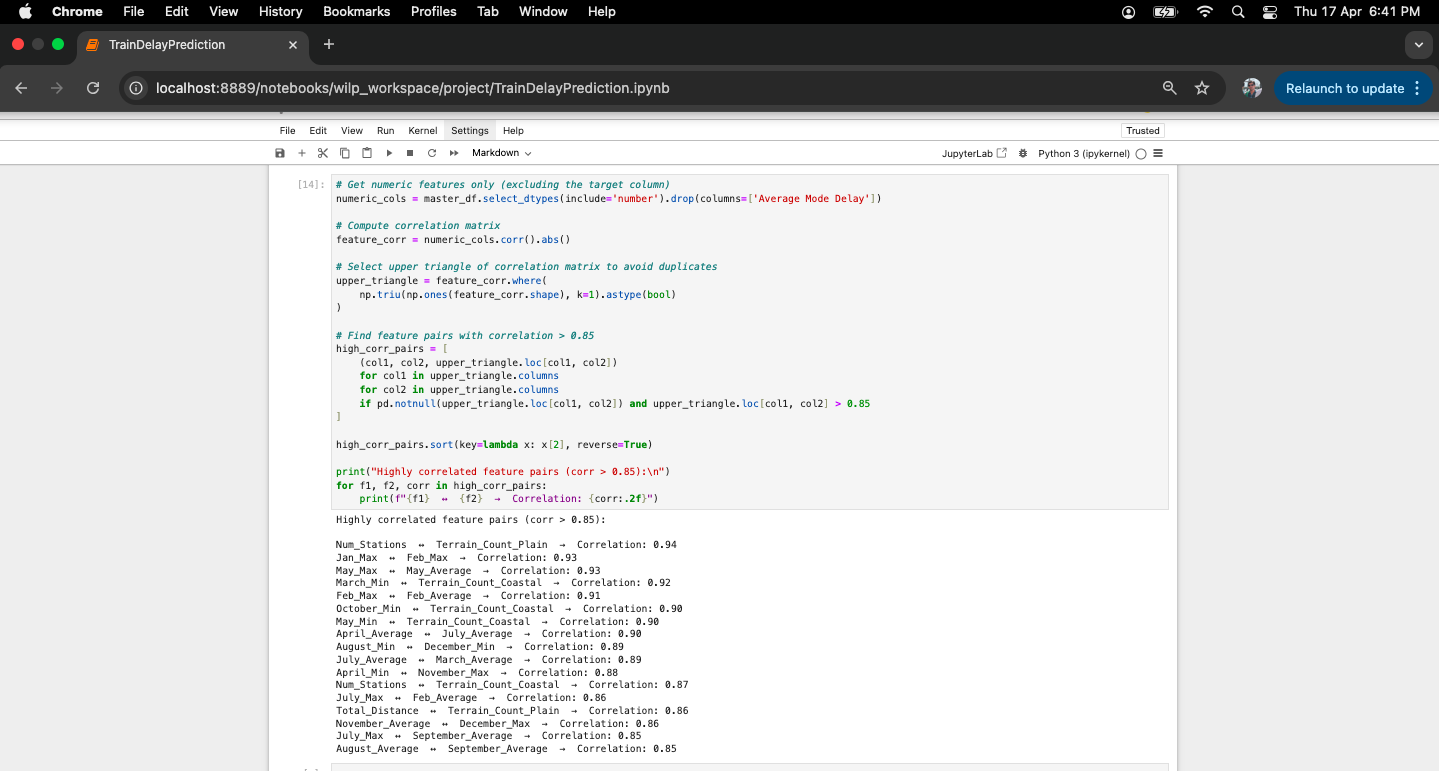
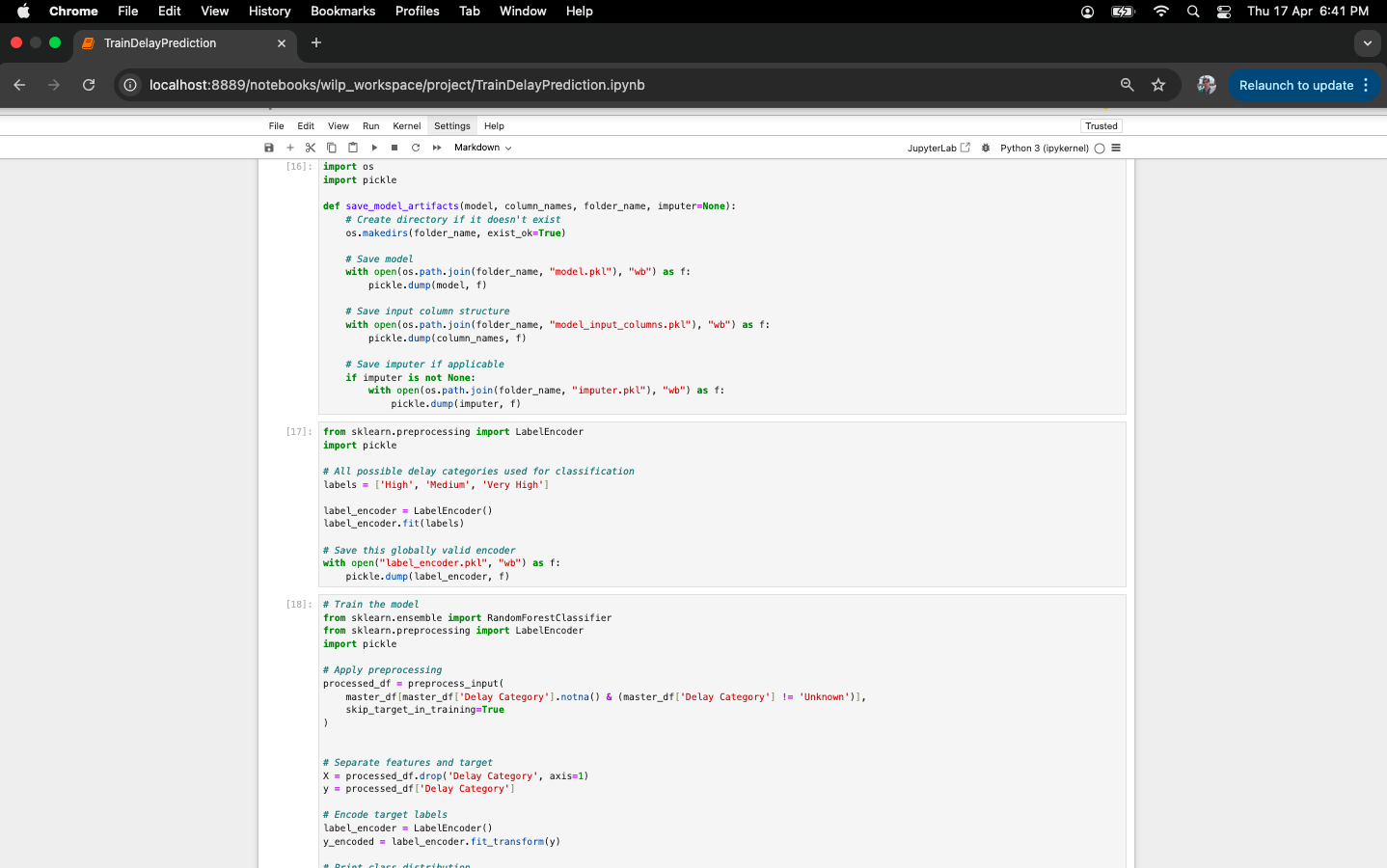
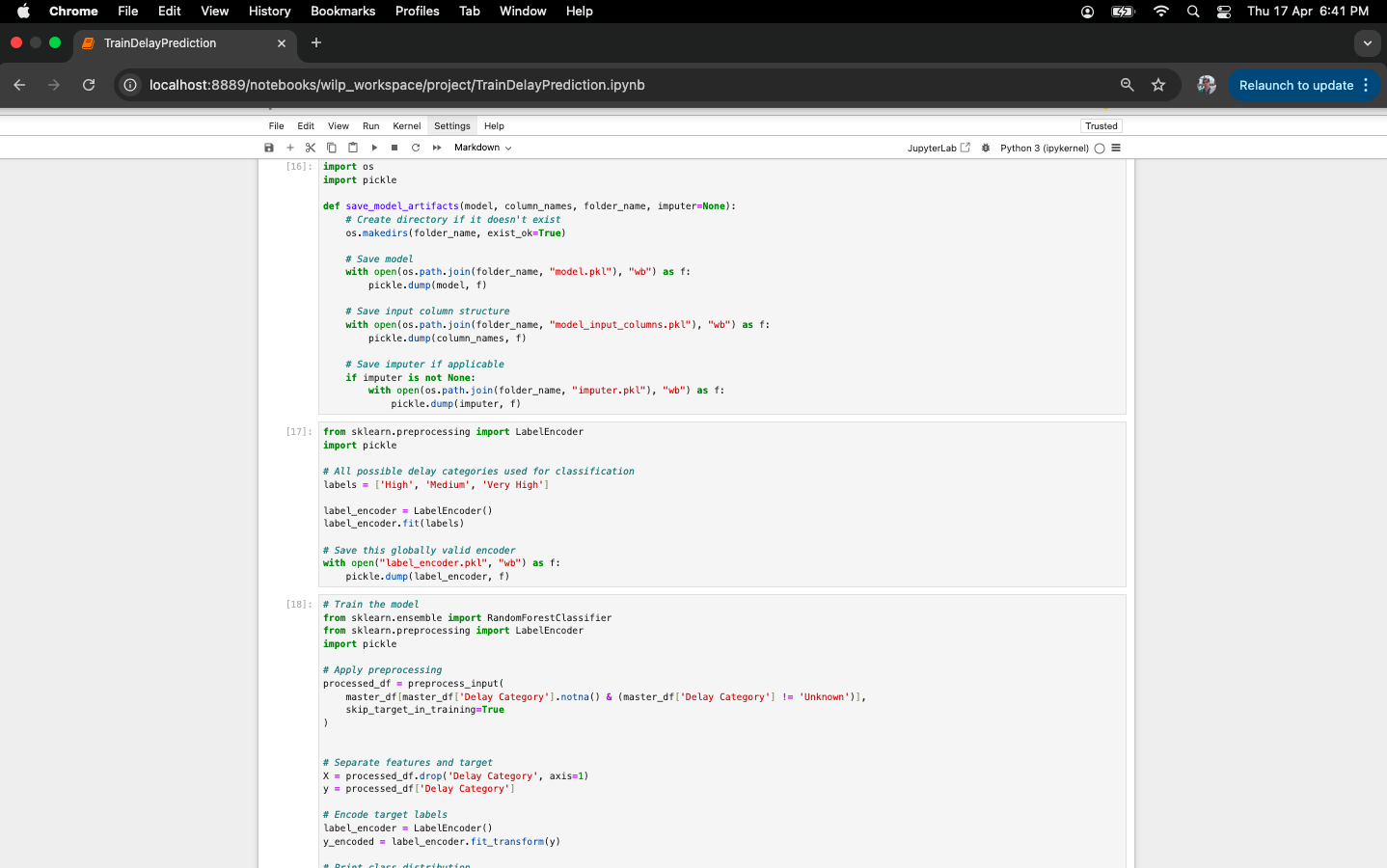
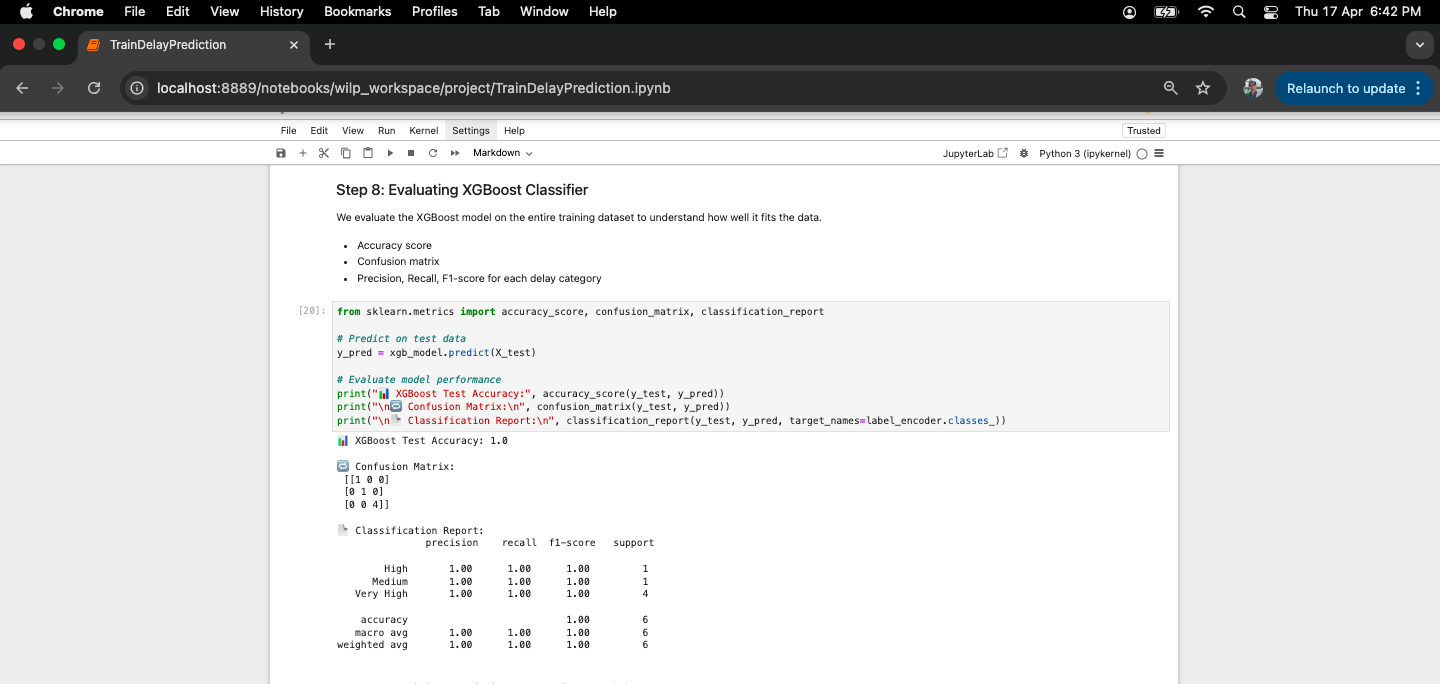
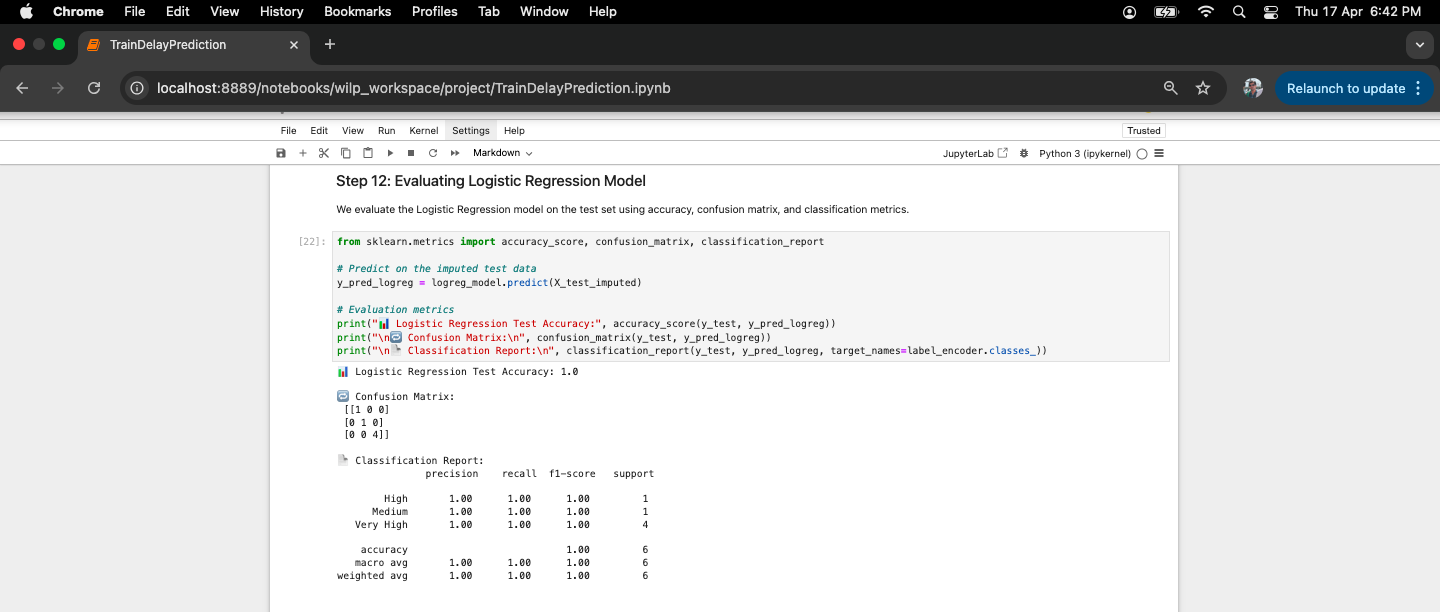
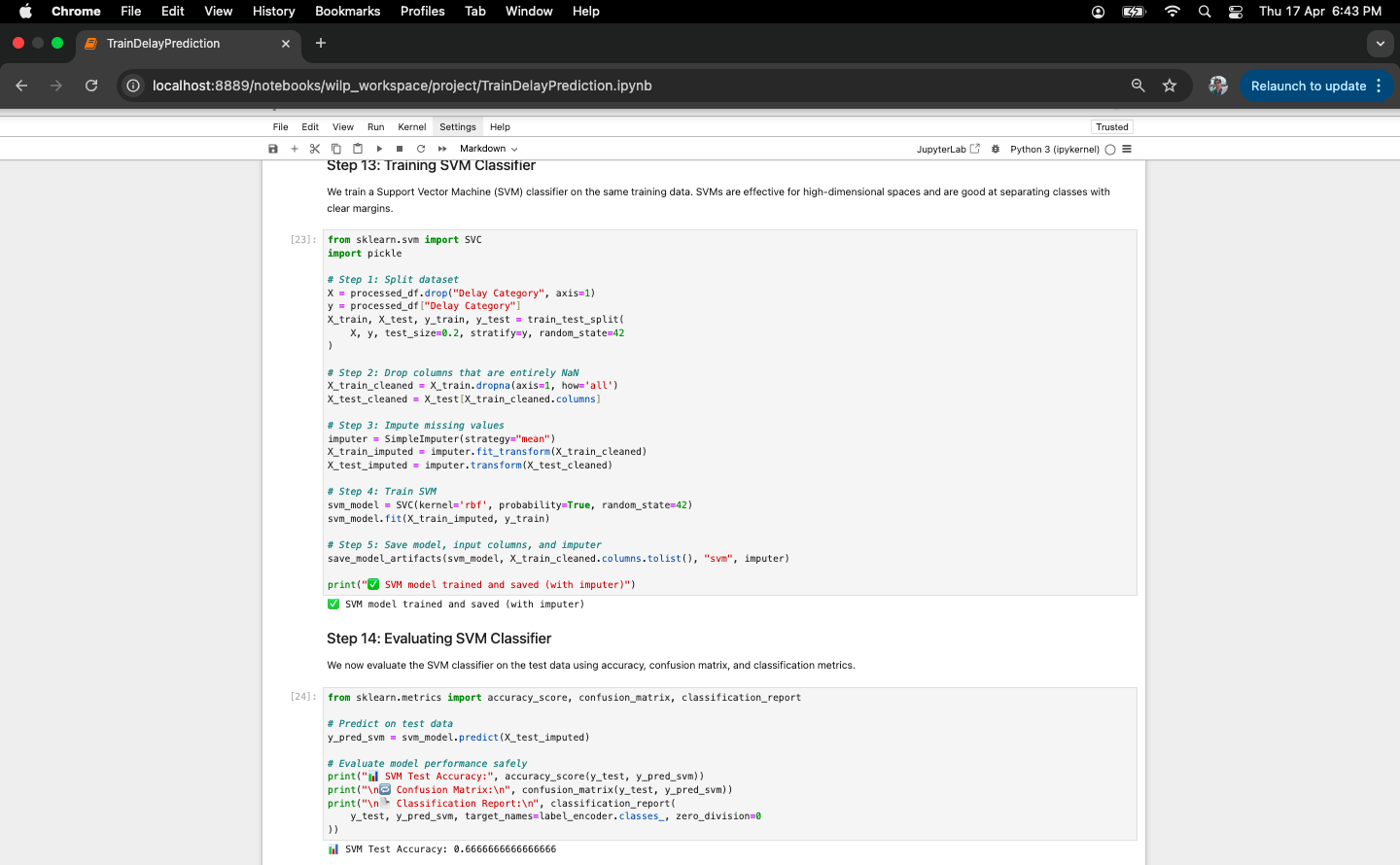
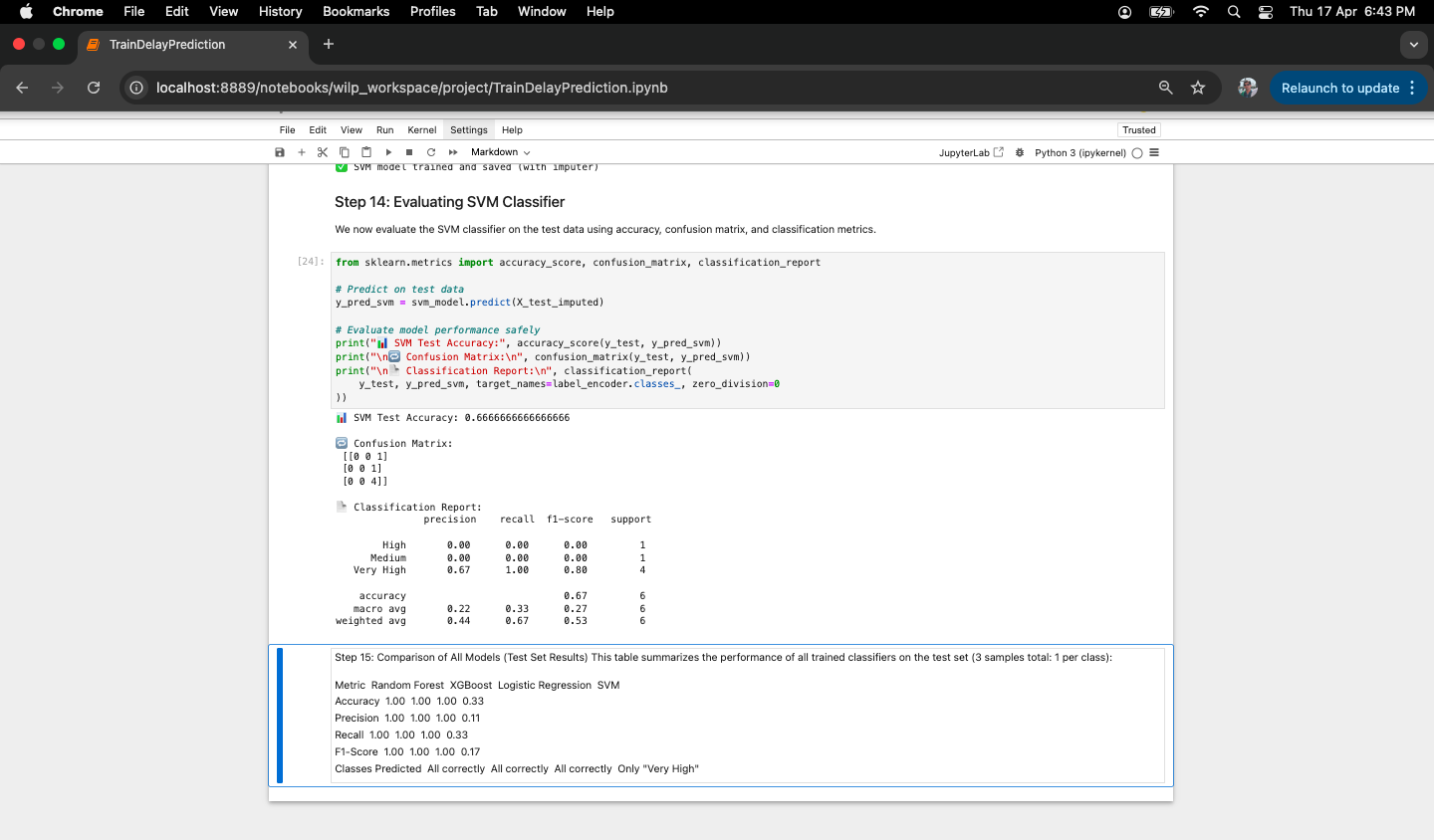
## **Training Details:**

• Features Used: Coach count, train type, station count, platform stats, terrain, travel time, days of run, departure/arrival time bins, etc.  
• Target Variable: Delay Category (derived from monthly mode delay)  
• Train-Test Split: 80% training, 20% testing, stratified by delay category  
• Missing Value Handling:  
 - Random Forest/XGBoost: NaNs handled natively  
 - Logistic Regression/SVM: Used SimpleImputer(strategy='mean') before training  
• Model Artifacts Saved:  
 - model.pkl – Trained model  
 - model\_input\_columns.pkl – Columns used during training  
 - imputer.pkl – Only for Logistic Regression and SVM  
• Each model was saved inside its own folder: random\_forest/, xgboost/, logistic\_regression/, svm/

**Model Evaluation Summary:**



## **Code & Screenshots**

1. Github repo : <https://github.com/ankitprjts/bitsGrp15Project>
2. Screenshots from Notebook:
   1. 
   2. 
   3. 
   4. 
   5. 
   6. 
   7. 
   8. 
   9. 
   10. 
   11. 
   12. 
   13. 
   14. 
   15. 
   16. 
   17. 
   18. 
   19. 
   20. 

## 

## **Deployment**

## **Objective:**

The deployment objective was to make the trained machine learning models accessible through an interactive and user-friendly web interface that allows users to input train details and get predicted delay categories in real time.

## **Tools and Technologies Used:**

1. Python: For building the entire pipeline including preprocessing, modeling, and inference logic
2. Streamlit: For developing the interactive web interface
3. Pickle: For saving and loading trained models, imputers, and encoders
4. Pandas & NumPy: For data manipulation
5. Scikit-learn & XGBoost: For training and predicting using ML models

## **Deployment Architecture Overview:**

The deployment architecture is designed to be modular and scalable. It uses a central Streamlit app (`app.py`) placed at the project root, and each trained model is stored in a dedicated directory. Each model directory includes the trained model file (`model.pkl`), input columns used for training (`model\_input\_columns.pkl`), and imputer (if needed).

When a user selects a model and submits input values, the app dynamically loads the corresponding model, applies the preprocessing logic (via `TrainDelayPrediction.py`), ensures that the feature columns match, applies imputation if required, and finally predicts the delay category.

## **Project Folder Structure:**

The following directory structure is used to organize the project:  
  
📁 project-root/  
│  
├── app.py  
├── label\_encoder.pkl  
├── TrainDelayPrediction.py  
├── random\_forest/  
│ ├── model.pkl  
│ └── model\_input\_columns.pkl  
├── xgboost/  
│ ├── model.pkl  
│ └── model\_input\_columns.pkl  
├── logistic\_regression/  
│ ├── model.pkl  
│ ├── model\_input\_columns.pkl  
│ └── imputer.pkl  
├── svm/  
 ├── model.pkl  
 ├── model\_input\_columns.pkl  
 └── imputer.pkl

## **Key Insights & Impactful/Important Features**

## **Key Insights**

1. The majority of trains experience low to moderate delays, with a few outliers that face very high delays.
2. Certain months show higher delay variability, indicating seasonal or operational patterns.
3. Routes with a higher number of stations and varied terrain are more prone to delays.
4. Trains with pantry facilities and higher coach counts do not necessarily correlate with higher delays, challenging some operational assumptions.
5. Some zones like NR and SCR exhibited higher variance in delays, possibly due to traffic congestion or regional infrastructure.

## **Important Features Identified**

1. Average Mode Delay: Aggregated target feature used for classification.
2. Travel Time: Derived from departure and arrival time; a critical feature for distinguishing long vs short routes.
3. Number of Stations: Higher values typically associate with more potential delays.
4. Platform Count (Avg/Min/Max): Proxy for infrastructure capacity; influenced route efficiency.
5. Train Type & Zone: Captures operational style and geographic spread; both show strong correlation with delays.
6. Departure and Arrival Time Bins: Helped capture time-of-day effects like peak hours or night-time buffers.
7. Terrain Categories: Presence of hills or coastal terrain introduces delay risk due to geographical constraints.

## **Future Work / Scope of improvements**

The current model for estimating train delays in Indian Railways, based on one year of historical data, provides valuable insights into delay patterns. However, there are several areas where the project can be further improved to increase its accuracy, scalability, and applicability. Below are some potential directions for future work:

#### **1. Expansion of Historical Data**

Currently, the model is trained on **one year of historical data** due to limitations in data access from the available websites (etrain.info and indianrail.gov). As there are no public APIs or access to data spanning more than a year, **future work** could focus on:

* **Exploring alternative data sources**: Working with Indian Railways directly or collaborating with research institutions for access to older data spanning **2–3 years**.
* **Using synthetic data** or **simulation techniques** to expand the training set and better account for long-term trends in train delays.

This will allow the model to capture **long-term seasonality**, **infrastructure changes**, and **systemic delays**, offering a more comprehensive view of the problem.

#### **2. Incorporating More Features**

The current model uses basic features like **Train details, route details, terrains, days of run etc.** Additional external factors that could be incorporated include:

* **Track maintenance schedules**: To understand how track repairs or construction might impact delays.
* **Previous day’s delay patterns**: Integrating delays from the **previous day** or historical data could help predict delays more accurately by capturing trends over time.
* **Passenger load and congestion**: Including data on passenger volume or train occupancy might offer insights into delays due to overcrowding or operational inefficiencies.

**3. Admin Dashboard for Ops**

Build a multi-tab Streamlit app that shows:

* Monthly delay trends
* Worst-affected routes
* Accuracy of models over time

**Impact:** Valuable for rail ops teams, not just passengers

**3. Smart Alerts / Notification System**

* Allow user to subscribe to a train ID and get predictions emailed/pushed before travel
* **Impact:** Improves user engagement and real-world utility

## **BIBLIOGRAPHY**

1. Reuters. (2025, January 8). *Indian Railways to spend over $22 billion on modernisation*. Reuters.<https://www.reuters.com/business/autos-transportation/indian-railways-spend-over-22-billion-modernisation-2025-01-08>
2. eTrain. (n.d.). *Train Running Status and Information Portal*. Data for the period April 2024 to March 2025 collected between March 3 and March 21, 2025, from<https://etrain.info/in>
3. Indian Railways. (n.d.). *Indian Railways Official Website*. Data for the period April 2024 to March 2025 collected between March 3 and March 21, 2025, from<https://indianrail.gov.in>
4. Gaurav, R. (n.d.). *Train Delay Estimation – Tutorial* [GitHub repository]. Retrieved March 3, 2025, from<https://github.com/R-Gaurav/train-delay-estimation/blob/master/doc/Tutorial.md>

## **Checklist of items for the Final report**

1. Is the Cover page in proper format? Y
2. Is the Title page in proper format? Y
3. Is the Certificate from the Mentor in proper format? Has it been signed? Y / N
4. Is Abstract included in the Report? Is it properly written? Y
5. Does the Table of Contents page include chapter page numbers? Y
6. Does the Report contain a summary of the literature survey? N
   1. Are the Pages numbered properly? Y
   2. Are the Figures numbered properly? Y
   3. Are the Tables numbered properly? Y
   4. Are the Captions for the Figures and Tables proper? Y
   5. Are the Appendices numbered? N
      1. Does the Report have Conclusion / Recommendations of the work? Y
      2. Are References/Bibliography given in the Report? Y / N
      3. Have the References been cited in the Report? Y / N
      4. Is the citation of References / Bibliography in proper format? Y / N